TMA 02

TT284: TMA 02

How to submit your assignment

This module uses the electronic TMA (eTMA) system for submission of TMAs. To submit your TMA, please go to your StudentHome page and follow the link(s) provided.

It is also helpful from your tutor’s point of view if you include your name and your personal identifier in the header of your document.

If you foresee any difficulty with submitting your assignment on time then you should contact your tutor well in advance of the cut-off date.

This tutor-marked assignment covers all parts of Block 2. It will make up a third of your continuous assessment score for this module. For more details about how your score for this module is calculated, see the ‘Breakdown of marks’ section of the Module Guide.

For further information about policy, procedure and general submission of assignments, please refer to the Assessment Policies. These can also be accessed via your StudentHome page.

Plagiarism

It is imperative that you avoid plagiarism as it is a form of cheating. If you do plagiarise then you won’t simply lose marks; instead, your work will be referred for further disciplinary action. If you are in any doubt as to what constitutes plagiarism, please refer to the University guidelines or see the 'Plagiarism’ section of the Module Guide.

Please also note that, unless directed otherwise, the use of automated text and code generation tools (such as ChatGPT, Bing Chat, or any similar) also falls under the definition of Plagiarism, as you would be passing off work that you have not done yourself as your own.

Tutor-marked assignment 02 (TMA 02)

Introduction to the assignment

For your assessment for this block, you are required to submit a tutor-marked assignment (TT284 TMA 02), which must be submitted by **12 noon (UK local time)** on the cut-off date: **1 February 2024**.

The aim of this assessment is to allow you to demonstrate your skills in understanding and extending the client-side and server-side functionality of an existing web application. This will allow you to demonstrate your progress towards achieving the block’s learning outcomes.

This assignment consists of two parts. In the first part, you will undertake some practical development work. In the second part, you will write a report covering topics from this block and reflecting on your implementation work.

Please read the guidance in the TMA questions and the ‘Assessment’ section of the Module Guide before answering the questions here. If you need further guidance on any assignment questions, please ask your tutor, who will be happy to help, or consult the TMA 02 forum. However, do not post any part of your assessed work to any forum.

This TMA is expected to take you about 10 hours; this estimate assumes that you have studied and completed the practical activities and exercises in the block before attempting the TMA.

It is better to attempt all parts of all questions. Limit the time you spend on each part before moving on. As far as possible, in the practical sections, you can complete later parts even if you do not reach a working solution for an earlier part. You can earn marks for incomplete solutions if you explain what you have done.

Background

In Block 2 Part 6 you met an example web application which collected user data, wrote it to a database and allowed it to be edited. Mike was quite impressed by that and would like it to be developed further. That application worked with first name, last name and email address, but Mike wants an additional data field.

The task here is to extend the code in the accompanying files in a similar way to the practical activity and solution in Block 2 Part 6. **You should work through Block 2 Part 6 before you attempt this question.**

The work here is based on the accompanying files which are similar to the solution given to Block 2 Part 6.

> Note that if you have not created the ‘tt284_guests’ table when completing the activities in Block 2 Part 6, or if you have subsequently deleted the ‘tt284_guests’ table, you will not be able to complete the tasks in this assignment in the order they are presented. You should create that table before you proceed.

You must work with the files provided with this assignment and edit only the files which you are explicitly asked to change.

Part 1: A web app for the Erehwon Guest House

Important notes

By default, browsers will cache the content they receive. That will lead you to making changes which don’t appear to work and waste your time. To avoid that, open the developer tools in your web browser, select the ‘Network’ tab and check the ‘Disable cache’ checkbox. Leave the developer tools open. These instructions are for Google Chrome, other browsers will vary. We recommend that you use Chrome.

Working with code can be very time consuming and you need to manage your time carefully. If you are not making good progress, step away from the task and return later.

(a) Connect to the database

Modify **credentials.php** to connect to your database.

(2 marks)

The PHP files will need access to the database provided on your server. Your credentials can be found on your server home page and should be entered in the file **credentials.php**

Your server home page can be found at https://*your_oucu*.tt284.open.ac.uk/.

Once you have done that, execute the file **table-manager.php** to verify that you can connect to the database.

> In your solution document, provide your code (see Appendix A) and a screenshot (see Appendix B) of the successful connection report as evidence of success.

(b) Client-side

i. **Update the page title and heading**

   Modify **tma02_admin.php** to update the main page title and heading.

   (2 marks)

Mike wants this development version to be clearly identified so he knows who is working on it. To clearly identify the page, change the existing text so that it shows your OUCU, followed by a dash and the text ‘Erehwon Guest House’.

For example, the student Zennata Quillington has an OUCU zxy999 so the heading will read:

zxy999 - Erehwon Guest House

and the title (shown in your web browser title or tab bar) will read:

zxy999 - Erehwon Guest House

Examine the code carefully and make the minimum number of changes necessary.

If you don't know what your OUCU is, see Appendix C.

> In your solution document, provide your code and a screenshot of the web page showing **both** the new title and page heading.

ii. **Adding style**

Modify **tma02_admin.css** to add the following style changes.

(5 marks)

Mike is concerned that there are some aspects of the look and feel which don't reflect his expectations. Despite it being a prototype, he is insistent that these be addressed.

He would ideally like the text to be in the 'Helvetica' font, but understands that cannot be guaranteed. Failing that he would like 'Arial', but if that is not possible, he wants to be sure that the font is from an appropriate generic family.

Modify one place in the CSS to use an appropriate font for the page text (note that text in the input fields and the buttons may not be changed).

Some elements are to be changed to less intense background colours. Mike isn't specific about the exact colour, so anything meeting his specifications will be fine.

The elements you need to change are:

- text inputs to have a pale green background
- valid feedback to be shown as bold black text on a yellow background
- invalid feedback to be shown as bold black text on a red background.

Change the text and background colours and text formatting as required. You may use any CSS mechanism to specify the colour. Note that autocomplete features in your web browser may affect the background colour of some inputs – you are not required to address this.

> In your solution document, provide your code **and** a screenshot of the web page showing the new colour scheme and font.

iii. **Add ‘booking reference’ field to the data entry form**

Modify **tma02_data_form.php** to provide a labelled text field to accept a string of 10 characters in the data entry form.

(4 marks)

Mike wants to also capture the booking reference in the form.

Add a label with the text ‘Booking Reference’ and a corresponding text input field, and feedback element to the data entry form. Note that the input field should have a size limit of 10 characters and must be completed with 10 characters by the user. Mike does not want to capture information other than from customers, so every entry will have a booking reference. When the limit is reached, the input field should stop accepting characters, but no message should be produced. Your new form field should have a suitable and meaningful name attribute and use the same name as its id.

The field value should be empty when the page loads.

Hint: If you examine the supplied code carefully and observe how the current labels, fields and feedback are implemented, these will give you a good framework from which you can create the additional label, field and feedback element.

In your solution document, provide your code **and** a screenshot of the web page showing the updated data entry form. Note that at this point the displayed field will not generate the validation messages that the other fields do.

iv. **Client-side validation**

Modify **tma02_admin.js** to provide client-side validation of the ‘Booking Reference’ field using JavaScript.

(5 marks)

Mike wants the client-side to check for mistyped booking references and report them rather than submitting them to the server. He has decided that JavaScript client-side validation is needed to achieve that.

All booking references start with a group of three uppercase letters, followed by a forward slash (/) and then a six-digit number..

You need to extend the existing JavaScript validation so that the ‘booking reference’ field is constrained as described.

To do that, you will need to add additional code to the function `validate()` to recognise and check the new field, and modify the function `validateForm()` so the form cannot be submitted without a valid booking reference.

Add a comment to your validation code explaining the meaning of each part of your booking reference validation expression.

Note: achieving this validation by other means will not gain any marks.

> In your solution document, provide your code **and** two screenshots, one showing the response when a valid booking reference is entered and the other when it is invalid.

(c) Server-side

i. **Create a database table**

   Modify **table-manager.php** to create a new table in your database.

   (3 marks)

So far the code has used an existing database table called tt284_guests, but that has no provision for a ‘booking reference’ field. You will now create a new table in your database.

Examine the file **table-manager.php**

Notice the variable `$database_table` which must be set to the name of your new table required. You will create a table called ‘tt284_oucu’. (Where 'oucu' is replaced with your own OUCU).

Your new table will contain all the columns which exist in tt284_guests, with the addition of the new ‘booking reference’ column. Base your answer closely on the existing SQL statement.

The ‘booking reference’ field may contain up to 10 characters.

Modify the SQL statement in the `$create_sql` variable to meet these requirements. Remember that column names cannot contain spaces, so you must choose a suitable name. Once you are happy with your SQL, open **table-manager.php** on the TT284 server and click the ‘create’ button to execute the SQL to create the table.

Once a table is created it cannot be modified, so if you create a table with the incorrect structure, you must delete it before you can create a corrected version. This can be done using the ‘drop’ button when executing **table-manager.php**.

In your solution document, provide your code and a screenshot of your browser displaying the result of executing **table-manager.php**

ii. **Configure to use the new table**

Modify **tma02_admin.php** to use the new table.

(2 marks)

Only a single line of code should be changed.

In your solution document, provide your line of code.

You should perform some tests to be sure that your code is using the new table before moving on, but you do not need to provide evidence of those tests.

iii. **Store ‘booking reference’ in the database**

Modify **tma02_save-row.php** to add the data from the ‘booking reference’ field to the database.

(8 marks)

Look carefully at how the other data elements are added to the database and reflect that for the ‘booking reference’ field.

Note carefully that **tma02_save-row.php** handles two distinct situations: one when the record has been edited and the other when the record is new. Here, we are only considering a new record and you need to update only that part, leaving the other unchanged.

Add the following two entries:

Table 1: Data to add to database

First name	Last name	Email	Booking reference

Zinolla	Zanda	z.zanda@zinc.ac.uk	ONL/135791
Zeua	Avrenim	zeua.avrenim@zmail.ac.uk	EGH/246810

If you have created any other records when testing, they should be removed prior to adding the two entries above. The quickest way to clear the test data is to drop and recreate the table using **table-manager.php**.

Table 2: Database table display

firstname	lastname	email
Zinolla	Zanda	z.zanda@zinc.ac.uk
Zeua	Avrenim	zeua.avrenim@zmail.ac.uk

Table 2 shows the data table which should now be displayed. Notice that the new ‘booking reference’ column is not currently shown.

In your solution document, provide your code and a screenshot of the web page showing the new table containing entries with the fields completed. You can use the ‘Show/Hide: Data from database table’ to produce the screenshot if required.

iv. **Display ‘booking reference’ column**

Modify **tma02_data-table.php** to display the ‘booking reference’ column using data from the database.

(4 marks)

You will need to display the ‘booking reference’ column from the database in the same way the name and email columns are displayed. You are not required to add the ‘booking reference’ column to the search feature, but there is no penalty for doing so.

The data table should now be as shown in Table 3 (your choice of booking reference column name may vary):

Table 3: Database table display with Booking Reference column

firstname	lastname	email	reference

Zinolla	Zanda	z.zanda@zinc.ac.uk	ONL/135791
Zeua	Avrenim	zeua.avrenim@zmail.ac.uk	EGH/246810

In your solution document, provide your code and a screenshot of the web page showing the table containing the new entries.

v. **Server-side Validation**

Modify **tma02_validate.php** to provide server-side validation for ‘booking reference’ field.

(8 marks)

Although the client-side validation checks the correct form of the booking reference, it does not actually check it is a valid booking reference.

For a reference to be valid, the server-side should validate the reference to the same format as the client-side, plus the following additional constraints: the three-letter group must be one of ““EGH”, “ONL”, “DSC”. Anything else is invalid.

The next character must be a forward slash (/). Anything else is invalid.

The first digit may only be 1 or 2. The other digits may be any value between 0 and 9 inclusive.

You need to provide checks and only if all are valid, proceed to write the data to the database. If not, provide feedback to the user that the booking reference is invalid.

Add a comment to your validation code explaining the meaning of each part of your booking reference validation expression.

Again, examine the existing code closely, it will guide you.

The easiest way to test regular expressions is using a free online tool – search the web for ‘regex tester’ to discover these.

In your solution document, provide a copy of your code.

(d) Validate your HTML

Validate the output of your final code using the W3C HTML validator.

(2 marks)

This task must be completed using the final code you have left on the server. You can gain marks in this task even if you did not complete the tasks in parts (b) and (c).

You will need to view **tma02_admin.php** in your browser (not using the developer tools) so that the HTML is generated, then view source. Copy the source code and then paste this to the validator using the option to validate by direct input. Make sure you are using the HTML5 and 'Show Source' options.

There should be no errors reported. If the generated HTML shows any warnings, you are not required to address these.

In your solution document, provide a screenshot of the part of the HTML validator result page from the Message Filtering button to the bottom of the page taking care that the source code at the bottom of the page shows the tags within the <head> tag of your source code. To reduce the height of this image, you may use the Message Filtering feature to 'Hide all warnings' but do not hide errors..

Part 2: Erehwon report

Mike wants a report to allow him to better understand web architecture and security (Parts 2(a) and 2(b)). Then, we want you to reflect on your work in this TMA and write about your experiences (Part 2(c)).

Each section has an individual focus and word limit, as specified below. Details on the TT284 word count policy and referencing can be found in the Module Guide.

The main body of the report should consist of three sections. You should introduce each section with a heading.

(a) Conducting research and recommending appropriate steps

The Erehwon guest house currently consists of seven rooms, but Mike has expansion plans and expects to be able to purchase a second guest house in the next few months. He then plans to buy further properties to expand to a chain of 20 guest houses (approximately 200 bedrooms) within the next 4 years. You should take that into account when answering this part.

Eventually, the website (a *web application*) will consist of the promotional pages you would expect for an organisation of this size, together with applications to:

- capture customer details
- allow booking records to be maintained
- allow customers to check room availability and book online
- allow customers to post reviews.

Mike wants to protect his investment by selecting a solution based on web technologies that will better withstand the test of time.

For this part of your report, you will first need to conduct some research using the module materials as jumping off points. You may find the following sections of Part 1 of the Block 2 module materials to be particularly helpful in shaping your research:

Section 3 Architecture

Section 5 Network and distributed architectures

Section 6 Realising an architecture

Section 8 Reflecting on architecture.

We want you to *identify* the requirements (current and future) plus four further aspects that need to be considered during this selection process.

For each aspect, *discuss* what needs to be considered to ensure that Mike's investment is protected over the next 4–6 years.

Then, *give your recommendations* for a web technology solution that will satisfy all of these aspects.

For each recommendation, *describe* how it helps to future-proof the solution and *justify* your answers with specific technological reasons related to the scenario.

The absolute word count for this part of your report is **600 words.**

Marks will be awarded based on the relevance of your research, the level of detail provided and inclusion of, at least, three separate good quality external references properly cited in the text and included in the reference list at the end of your report.

(**30 marks** for this section of your report)

(b) Discuss existing security features and describe possible

enhancements

The second topic concerns security. Mike has read a lot about cyber-attacks and needs advice on the security of the website . For this question, you should exclude the fact that the TT284 web server requires an Open University login, and assume the web application is open to all visitors.

First, he wants an explanation of which aspects of the existing code are helping to keeping it secure.

Choose three aspects of the current code to write about, and in your report describe each of these aspects and explain how it improves security, and reduces the risks to Mike’s business of not including each of them.

Mike then wants you to advise on how to improve security of this website even further.

Think of two possible ways to further improve security of the application (at least one of which should be based on information you have found outside of the module materials), and in your report describe each of these enhancements, and explain the risks to Mike’s business of not implementing them. These enhancements could be further developments of the aspects stated above or completely new. In this section, you may also consider any security measures which could be applied to the hosting platform.

Marks will be awarded based on the relevance of your content, the level of detail provided and inclusion of relevant external references properly cited in the text and included in the reference list at the end of your report.

The absolute word count for this part of your report is 500 words.

(**15 marks** for this section of your report).

(c) Reflection

In this section of your report, honestly reflect on the practical development work you undertook in Part 1. In particular, you must cover the following three questions:

- What went well in your development process? If nothing went well, tell us why you think this was.
- What went wrong in your development process? If nothing went wrong, tell us why you think this was.
- What did you learn that could be used to improve your future development process?

Two things are important to keep in mind for a good reflection. First, the aim is to analyse the strengths and weaknesses of the development process, not the strengths and weaknesses of your solution. This means that a weak solution can still have an excellent reflection of its development process. Second, the aim is to be honest and not to try to hide any weaknesses as well as to ensure that you can learn from the experience moving forward into the final TMA and the EMA.

The absolute word count for this part of the report is 300 words.

Marks will be awarded based on how well your reflection relates to the work submitted in Part 1.

(**5 marks** for this section of your report).

Finalising your report

Combine each of the sections described in Part 2 into a single report. The structure of a report is outlined in Appendix D and described in more detail below.

First, the report needs an appropriate and meaningful title, such that readers will clearly understand what it is about.

You must then provide an introduction that explains what the report contains.

Your introduction will be followed by the sections that you have already written. Each must have a heading and include the word count for that section.

You then need to add a conclusion that briefly summarises what the report has said.

Throughout the report remember that you are writing for a non-technical audience and that you must ensure that where you discuss technical concepts, you introduce these first. Also, remember that you should always relate the concepts you are discussing to the question context (the Erehwon Guest House).

Finally, you must include a reference list giving all the sources you used. Your written work is expected to draw from the module materials, but you must also use one or more external sources to support or justify one or more of your arguments as directed in the questions.

Each reference needs a citation in the text where you have used the ideas or quoted, and must be in the Cite Them Right (CTR) Harvard form. Your reference list will not be counted toward the word limit for the report.

> If you use automated tools and reference a website with no apparent author, many of them will use the title as the author. CTR Harvard also allows the organisation to be used and in many cases that will make a shorter citation and a clearer reference list. You may wish to manually adjust these (or contact the supplier and request that the tool behaviour is changed) as otherwise your intext citations can be very long.

You can find more information about referencing and plagiarism on the Library pages. There is also information available on how to cite module materials.

Marks will be awarded for a suitable title, headings, spelling, grammar, structure and referencing as well as content. Figures and tables should be properly captioned.

(**5 marks** for the overall structure of your report)

Total word count

The maximum length of your report cannot exceed 1600 words. This is an absolute limit, and markers will stop reading when the maximum word count is reached.

This limit is more than the sum of the sections, to allow for your introduction and conclusion (up to 200 words).

Likewise, the word limits stated for specific sections of your report are also maximums, so you must stay within the word count for each section.

More details on the TT284 word count policy can be found in the Module Guide.

What to submit to the eTMA system and TT284 server

All the code files for this TMA should be uploaded to your TMA 02 folder on the TT284 server before the cut-off date of the TMA.

Any files not needed for TMA 02 should be removed from the TMA 02 folder on the server. Take care when doing this as files once removed cannot be recovered! Keep backups of your work on your own computer and/or cloud storage throughout your work on TMA02.

Only your word-processed solution document containing your answers to both parts of the TMA should be submitted through the eTMA system. Clearly mark question numbers and in each part make sure you have provided the evidence requested to show you have completed the work.

The files on the server will be used to test your solution, but it is the code you provide in your solution document that will be marked. All your answers to the TMA 02 questions should be submitted as a single document. Please name the file using your own OUCU ‘OUCU_TT284_TMA02.doc’ or ‘OUCU_TT284_TMA02.docx’.

If you don’t know what your OUCU is, see Appendix C.

Breakdown of marks

Part 1 is worth 45% of the marks available for this assignment and Part 2 is worth 55%. These percentages are further broken down below:

Part 1 – marks will be awarded for:

(a) connecting to the database	2
(b) client-side	16
(c) server-side	25
(d) validating your HTML.	2

Total marks available for Part 1	**45**

Part 2 – marks will be awarded for:

(a) conducting research and recommending appropriate steps	30
(b) discussing existing security features and describing possible enhancements	15
(c) reflection covering what went well, what went wrong, and what you learned from the development process in Part 1	5
(d) report structure, style, grammar, spelling, punctuation and referencing.	5
Total marks available for Part 2	**55**

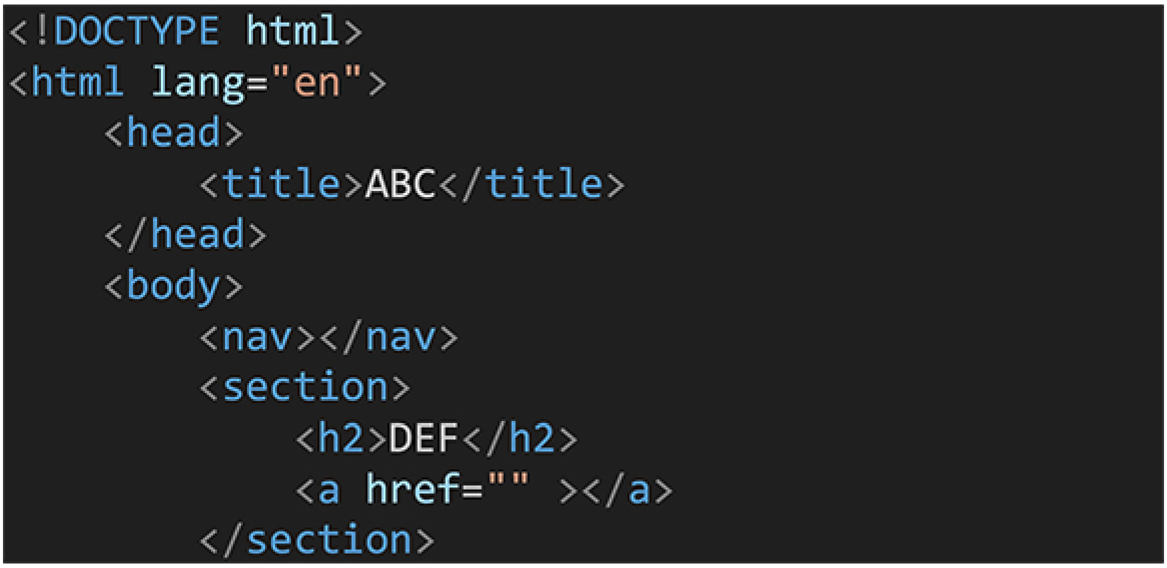
Appendices

Appendix A – Presenting your code

Code should be presented as text, not an image. Use a fixed pitch font such as `Courier New`.

You need to consider how much code to provide. If you change only a small part of a file, it is usually best to provide the changed lines together with a few lines from above and below, to give it context. Do not include the entirety of your code.

If you wish, you can use a tool such as Visual Studio Code (Microsoft, 2023) which allows you to simply copy and paste syntax-highlighted code into your solution document.



```
<!DOCTYPE html>
<html lang="en">
    <head>
        <title>ABC</title>
    </head>
    <body>
        <nav></nav>
        <section>
            <h2>DEF</h2>
            <a href="" ></a>
        </section>
```

Figure 1: Copy and pasted extract from Microsoft Visual Studio Code (Microsoft, 2023)

Reference

Microsoft (2023) *Visual Studio Code*. Available at: https://code.visualstudio.com/ (Accessed: 16 June 2023).

Appendix B – Presenting screenshots

Screenshots should contain only the necessary material and have a caption to explain their purpose.

You can either capture just what you want using the ‘Snipping Tool’ in Windows or ‘Grab’ on a Mac, or capture the whole screen then crop the result. Linux users can install products such as ‘Snapshot’.

Appendix C – Finding your OUCU

Your OUCU is your Open University computer username. It consists of a few letters followed by up to six numbers.

You can find your OUCU on your StudentHome page. Go to ‘Profile’ > ‘Update personal details’, then from ‘About you’, choose ‘Contact Details’ and it will be displayed.

If you have any difficulty finding your OUCU, the Computing Helpdesk will be able to help.

Appendix D – Report structure

A report must be written in language suitable for its intended audience (Mike) and have:

- a meaningful title
- a short introduction
- a main body containing your answers to Parts 2(a), 2(b) and 2(c), clearly divided into sections covering different aspects, including in-text citations matching the reference list
- a short conclusion
- a statement of the word count
- a reference list in the Cite Them Right Harvard style.